Kali Linux Wireless Penetration Testing Essentials

Kali Linux is the most popular distribution dedicated to penetration testing that includes a set of free, open-source tools.

This book introduces you to wireless penetration testing and describes how to conduct its various phases. After showing you how to install Kali Linux on your laptop, you will verify the requirements of the wireless adapter and configure it. Next, the book covers the wireless LAN reconnaissance phase, explains the WEP and WPA/WPA2 security protocols and demonstrates practical attacks against them using the tools provided in Kali Linux, Aircrack-ng in particular. You will then discover the advanced and latest attacks targeting access points and wireless clients and learn how to create a professionally written and effective report.

Who this book is written for
This book is targeted at information security professionals, penetration testers and network/system administrators who want to get started with wireless penetration testing. No prior experience with Kali Linux and wireless penetration testing is required, but familiarity with Linux and basic networking concepts is recommended.

What you will learn from this book
• Explore the penetration testing methodology and its various phases
• Install Kali Linux on your laptop and configure the wireless adapter
• Scan and enumerate wireless LANs and point out their vulnerabilities
• Understand the WEP security protocol and the techniques to crack the authentication keys and break it
• Become proficient with the WPA/WPA2 protocol and use Kali Linux tools to attack it
• Attack the access points and take control of the wireless network
• Launch advanced attacks against clients
• Produce stunning and effective reports

Plan and execute penetration tests on wireless networks with the Kali Linux distribution
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 3 'WLAN Reconnaissance'
- A synopsis of the book’s content
- More information on Kali Linux Wireless Penetration Testing Essentials
About the Author

Marco Alamanni has professional experience working as a Linux system administrator and information security administrator/analyst in banks and financial institutions.

He holds a BSc in computer science and an MSc in information security. His interests in information technology include, among other things, ethical hacking, digital forensics, malware analysis, Linux, and programming. He also collaborates with IT magazines to write articles about Linux and IT security.
Since their introduction to the market less than 20 years ago, wireless networks have grown exponentially and become ubiquitous, not only in the enterprises but everywhere else—all kinds of public places (coffee shops, restaurants, shopping malls, stations, and airports), open-air free Wi-Fi zones, and private homes.

Like all other technologies, their spread has led to a growing need for assessing and improving their security, as a vulnerable wireless network offers an easy way for an attacker to access and attack the whole network, as we will see through this book.

For these reasons, the process of the security assessment of wireless networks, also called wireless penetration testing, has become an essential part of more general network penetration testing.

In this book, we explore the whole process of performing wireless penetration tests with the renowned security distribution of Kali Linux, analyzing each phase, from the initial planning to the final reporting. We cover the basic theory of wireless security (protocols, vulnerabilities, and attacks) but mainly focus on the practical aspects, using the valuable, free, and open source tools provided by Kali Linux for wireless penetration testing.

**What this book covers**

*Chapter 1, Introduction to Wireless Penetration Testing*, presents the general concepts of penetration testing and covers its four main phases with a particular focus on wireless networks.

The chapter explains how to agree and plan a penetration test with the customer and gives a high-level view on the information collection, attack execution, and report writing phases of the process.


**Preface**

Chapter 2, *Setting Up Your Machine with Kali Linux*, introduces the Kali Linux distribution and the included tools that are specifically designed for wireless penetration testing. Then we see the hardware requirements for its installation, the different installation methods, and also cover, step by step, installation in a VirtualBox machine, supplying the relative screenshot for every step.

After installing Kali Linux, the chapter exposes the features that the wireless adapter must meet to be suitable for our purposes and how to practically test these requisites.

Chapter 3, *WLAN Reconnaissance*, discusses the discovery or information gathering phase of wireless penetration testing. It begins with the basic theory of the 802.11 standard and wireless local area networks (WLANs) and then covers the concept of wireless scanning that is the process of identifying and gathering information about wireless networks.

We then learn how to use the tools included in Kali Linux to perform wireless network scanning, showing practical examples.

Chapter 4, *WEP Cracking*, speaks about the WEP security protocol, analyzing its design, its vulnerabilities and the various attacks that have been developed against it.

The chapter illustrates how command-line tools and automated tools can be used to perform different variants of these attacks to crack the WEP key, demonstrating that WEP is an insecure protocol and should never be used!

Chapter 5, *WPA/WPA2 Cracking*, starts with the description of WPA/WPA2 cracking, its design and features, and shows that it is secure. We see that WPA can be susceptible to attacks only if weak keys are used. In this chapter, we cover the various tools to run brute force and dictionary attacks to crack WPA keys. Also, recent and effective techniques for WPA cracking such as GPU and cloud computing are covered.

Chapter 6, *Attacking Access Points and the Infrastructure*, covers attacks targeting WPA-Enterprise, access points, and the wired network infrastructure. It introduces WPA-Enterprise, the different authentication protocols it uses and explains how to identify them with a packet analyzer. Then, it covers the tools and techniques to crack the WPA-Enterprise key.

The other attacks covered in the chapter are the Denial of Service attack against access points, forcing the de-authentication of the connected clients, the rogue access point attack and the attack against the default authentication credentials of access points.
Chapter 7, Wireless Client Attacks, covers attacks targeting isolated wireless clients to recover the WEP and the WPA keys and illustrates how to set up a fake access point to impersonate a legitimate one and lure clients to connect to it (an Evil Twin attack). Once the client is connected to the fake access point, we show how to conduct the so-called Man-in-the-middle attacks using the tools available with Kali Linux.

Chapter 8, Reporting and Conclusions, discusses the last phase of a penetration test, which is the reporting phase, explaining its essential concepts and focusing, in particular, on the reasons and purposes of a professional and well-written report.

The chapter describes the stages of the report writing process, from its planning to its revision, and the typical professional report format.

Appendix, References, lists out all the references in a chapter-wise format. We also cover the main tools included in Kali Linux to document the findings of the penetration test.
In this chapter, we are going to introduce the basic concepts behind wireless LANs (Local Area Networks) and see how to carry out the reconnaissance and information gathering phase of our penetration test.

This chapter deals with wireless networks scanning and information gathering, enumerating visible and hidden networks, identifying the security protocols used, their possible vulnerabilities and the connected clients. The topics covered are as follows:

- Introduction to 802.11 standard and wireless LAN
- Introduction to wireless scanning
- Wireless scanning with airodump-ng
- Wireless scanning with Kismet

Introduction to 802.11 standard and wireless LAN

Before diving into the hands-on part, it is worth recalling the basic concepts of the 802.11 standard on which wireless local area networks are based.

The 802.11 is the second layer (link layer) standard for implementing wireless LAN developed by the IEEE. Devices and networks that use the 802.11 standard are commonly known as Wi-Fi, a trademark of the Wi-Fi Alliance.

There have been subsequent specifications of the standard over the time, the main ones are 802.11a, 802.11b, 802.11g, and 802.11n.
WLAN Reconnaissance

802.11a operates on the 5 GHz frequency range while 802.11b/g on the 2.4 GHz frequency range, which is by far the most used by Wi-Fi networks nowadays. 802.11n supports both these frequency bands and is backward compatible with the other 802.11 specifications.

The range of the Wi-Fi signal depends on the standard used, on the power of the transmitting device and on the presence of physical obstacles and radio interferences.

For common Wi-Fi devices, it typically varies from a maximum of 20-25 meters indoors to 100 meters and more outdoors.

The maximum throughput, that is, the maximum data rate, of the 802.11 standard varies from the 11 Mbps of the 802.11a/b standards to the 600 Mbps of the 802.11n standard.

Each frequency band is subdivided into multiple channels, which are subsets that include smaller frequency ranges. The 2.4 GHz band is subdivided into 14 distinct channels, but not all of them are always used. Most of the countries typically allow only a subset of these channels, while some countries allow all the channels.

For example, United States allows channels from 1 to 11, while Japan allows all 14 channels. Indeed, every country has established its own regulatory domain (regdomain), a set of rules that defines the radio spectrum allocation for wireless transmission. The regulatory domains also define the maximum transmit power allowed.

About Wi-Fi channels
To get more information about Wi-Fi channels and regulatory domains, refer to the resource on Wikipedia at https://en.wikipedia.org/wiki/List_of_WLAN_channels.

802.11 frames, types, and subtypes
A 802.11 frame is composed of the MAC header, Payload, and Frame Check Sequence (FCS) sections, as shown in the following diagram:
The MAC header section is divided into various fields, among which are the **Type** and **Subtype** fields. The 802.11 standard defines three different types of frames:

- **Management frames**: These frames coordinate communication between access points and clients on a wireless LAN. Management frames include the following subtypes:
  - **Beacon frames**: These are used to announce the presence and the basic configuration of an access point (AP).
  - **Probe request frames**: These are sent by the clients to test for the presence of APs or a specific AP to connect to.
  - **Probe response frames**: These are sent by the AP in response to probe requests, containing information about the network.
  - **Authentication request frames**: These are sent by clients to begin the authentication phase prior to connect to an AP.
  - **Authentication response frames**: These are sent by the AP to accept or reject the authentication of the client.
  - **Association request frames**: These are used by the client to associate with the AP. It must contain the SSID.
  - **Association response frames**: These are sent by the AP to accept or reject the association with the client.

- **Control frames**: They are used to control the flow of data traffic on the network. The subtypes of control frames are **Request-to-send (RTS)** frames and **Clear-to-send (CTS)** frames, which provide an optional mechanism to reduce frame collisions and **Acknowledgment (ACK)** frames that are sent by the receiving station to confirm the correct receipt of a data frame.

- **Data frames**: These contain the data transmitted over the network, with packets of higher-layer protocols encapsulated in the 802.11 frames.

In the next section, we are going to recall the structure and the building blocks of a wireless network.
Infrastructure mode and wireless access points

Wi-Fi networks use the 802.11 standard in infrastructure mode. In this mode, devices called access points (APs) are used to connect the wireless client stations with a wired LAN or with the Internet. Access points could be seen as the analogue of switches for wired networks but they offer more functionalities such as network layer routing, DHCP, NAT, and advanced management capabilities through the remote console or the web administration panel.

A wireless network formed by a single AP is called a Basic Service Set (BSS) while a network with multiple APs is known as an Extended Service Set (ESS). Each AP is identified by the Basic Service Set ID (BSSID), which typically corresponds to the MAC address of the wireless interface on the AP. A wireless LAN is instead identified by the Service Set ID (SSID) or Extended Service Set ID (ESSID), which is usually a readable string that is used as the name of the network.

Access points periodically send out broadcast beacon frames to announce their presence. Typically, the beacons also contain the SSID of the AP, so that it is easily identifiable by clients, which can send authentication and association requests to the AP, to connect to the wireless network.
Wireless security
Data transmission on wireless networks is inherently less secure compared to wired networks regarding the physical media, because anyone nearby could sniff the traffic easily. Wireless LANs can use open authentication, such as free Wi-Fi hotspots do, and in this case no authentication is required from the clients and the traffic is not encrypted, making open networks totally insecure.

Two security protocols that provide authentication and encryption to wireless LANs have been developed over the time: Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA/WPA2).

The WEP and WPA/WPA2 authentication protocols and their relative cracking techniques will be discussed in Chapter 4, WEP Cracking and Chapter 5, WPA/WPA2 Cracking, respectively.

Wireless LAN scanning
The process of thoroughly examining the radio waves to find wireless network is called wireless scanning.

Wireless networks scanning has become quite popular, even among nontechnical people, also due to the so-called wardriving phenomenon. Wardriving is the activity of pinpointing wireless networks outdoors, usually driving a car and equipped with a laptop, a high-gain antenna and a GPS receiver.

There are two main types of scanning: active and passive.

- Active scanning involves sending broadcast probe request packets and waiting for probe response packets from access points, taking note of the discovered ones. This is the standard method used by clients to identify wireless networks that are available nearby. The disadvantage of this method is that an access point can be configured to ignore the broadcast probe request packets and to exclude its SSID from the beacons it sends (hidden AP), so in this case, active scanning could not identify the network.
- Passive scanning provides better results in regard of wireless Reconnaissance and is the method adopted by wireless scanners. In passive scanning, we don't send broadcast probe requests. The wireless adapter is instead put in monitor mode so that it can sniff all the traffic going on a given channel of the Wi-Fi frequency range. The captured packets are analyzed to determine which access points are transmitting, from the BSSID contained in the beacons, and which clients are connected. This way, access points that are hidden from active scanning can also be revealed.
The tools for scanning wireless networks included in Kali Linux fall in the category of passive scanners. We cover the two most popular of these tools in this chapter, airodump-ng and Kismet, but also tools such as Fern Wi-Fi Cracker and Wiﬁte can be used for this purpose. In the upcoming subsection, we see how to configure our wireless adapter in monitor mode.

### Configuring the wireless adapter in monitor mode

In the previous chapter, we have seen how to put the wireless interface in monitor mode, to verify that it is compatible with packet sniffing. Now, we analyze the details of this procedure.

Recall that we issued the `airmon-ng start wlan0` command, as shown in the following screenshot:

![Screenshot of `airmon-ng start wlan0` command output]

The `airmon-ng` tool also indicates us the chipset and the driver in use by the adapter. Notice that the `mon0` interface is created with monitor mode enabled, while the `wlan0` interface is in managed mode (which is the default mode for wireless adapters), as shown in the following output of the `iwconfig` command:
The `mon0` interface is listening on all the channels. If we want to listen on a specific channel, we can issue the `airmon-ng start wlan0 <channel>` command:

We see that another interface named `mon1` has been created in monitor mode. We can create multiple monitor mode interfaces related to a physical wireless interface. While running `airmon-ng`, we notice a warning telling us that some processes may interfere with other tools of the `Aircrack-ng` suite. To stop these processes, we can execute `airmon-ng check kill`.

We see that another interface named `mon1` has been created in monitor mode. We can create multiple monitor mode interfaces related to a physical wireless interface. While running `airmon-ng`, we notice a warning telling us that some processes may interfere with other tools of the `Aircrack-ng` suite. To stop these processes, we can execute `airmon-ng check kill`.
WLAN Reconnaissance

If we want to stop the `mon0` interface, we run the `airmon-ng stop mon0` command:

```
root@kali:~# airmon-ng stop mon0
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>Chipset</th>
<th>Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>wlan0</td>
<td>Atheros AR9285</td>
<td>ath9k - [phy0]</td>
</tr>
<tr>
<td>mon0</td>
<td>Atheros AR9285</td>
<td>ath9k - [phy0]  (removed)</td>
</tr>
</tbody>
</table>

Now that the interface is in monitor mode, we can proceed with wireless scanning.

**Wireless scanning with airodump-ng**

The `airodump-ng` tool is one of the many tools included in the Aircrack-ng suite. It is capable of sniffing and capturing 802.11 frames, besides recording information relative to discovered access points and clients. Airodump-ng scans the Wi-Fi frequency band, hopping from one channel to another. To use it, after having put the wireless interface in monitor mode, as we saw previously, we run the `airodump-ng mon0` command. The following screenshot shows its output:
The first line shows the last association between an AP and a client, with the current channel, the elapsed running time, and the security protocol used. As we can notice in the preceding screenshot, the top half of the screen displays the APs while the bottom half displays the clients.

For each AP found, the following information is shown:

- The BSSID (MAC address)
- The Power Level (PWR) and the Receive Quality (RXQ) of the signal
- The number of beacons sent and the number of captured data packets
- The channel (CH)
- The maximum speed supported (MB)
- The encryption algorithm (ENC), the cipher (CIPHER), and the authentication protocol (AUTH) used
- The wireless network name or SSID (ESSID)

If `<length: number>` appears in the ESSID field, it means that the SSID is hidden and the AP only reveals its length (number of characters). If the number is 0 or 1, it means the AP does not reveal the actual length of the SSID.

<table>
<thead>
<tr>
<th>BSSID</th>
<th>Power Level</th>
<th>Beacons</th>
<th>#Data, #/s</th>
<th>CH</th>
<th>MB</th>
<th>ENC</th>
<th>CIPHER</th>
<th>AUTH</th>
<th>ESSID</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:7A:4C:3C:0C:E9</td>
<td>-56</td>
<td>595</td>
<td>404</td>
<td>0</td>
<td>0</td>
<td>WPA</td>
<td>CCMP</td>
<td>PSK</td>
<td>InfoStradaWiFi-Z</td>
</tr>
<tr>
<td>08:24:17:02:09:57</td>
<td>-74</td>
<td>731</td>
<td>45</td>
<td>0</td>
<td>5</td>
<td>WPA</td>
<td>CCMP</td>
<td>PSK</td>
<td>TISCALI</td>
</tr>
<tr>
<td>28:92:4A:62:53:67</td>
<td>-74</td>
<td>672</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>WPA</td>
<td>CCMP</td>
<td>PSK</td>
<td>TP-LINK_7FEA4</td>
</tr>
<tr>
<td>08:7A:3C:0C:E9</td>
<td>-56</td>
<td>595</td>
<td>404</td>
<td>0</td>
<td>0</td>
<td>WPA</td>
<td>CCMP</td>
<td>PSK</td>
<td>AirPort d1 5A</td>
</tr>
<tr>
<td>08:24:17:02:09:57</td>
<td>-74</td>
<td>731</td>
<td>45</td>
<td>0</td>
<td>5</td>
<td>WPA</td>
<td>CCMP</td>
<td>PSK</td>
<td>HP_Network_Lan-o</td>
</tr>
<tr>
<td>08:7A:3C:0C:E9</td>
<td>-56</td>
<td>595</td>
<td>404</td>
<td>0</td>
<td>0</td>
<td>WPA</td>
<td>CCMP</td>
<td>PSK</td>
<td>HP_Network_Lan-o</td>
</tr>
<tr>
<td>08:7A:3C:0C:E9</td>
<td>-56</td>
<td>595</td>
<td>404</td>
<td>0</td>
<td>0</td>
<td>WPA</td>
<td>CCMP</td>
<td>PSK</td>
<td>TISCALI</td>
</tr>
</tbody>
</table>

If `<length: number>` appears in the ESSID field, it means that the SSID is hidden and the AP only reveals its length (number of characters). If the number is 0 or 1, it means the AP does not reveal the actual length of the SSID.
WLAN Reconnaissance

In the bottom half, the \textit{STATION} field is about the MAC address of the clients, which can be associated with an AP. If associated, the BSSID of the AP is shown in the relative field; otherwise, the \textit{not associated} state is displayed.

The \textit{Probes} field indicates the SSIDs of the APs the client is trying to connect to, if it is not currently associated. This can reveal a hidden AP when it responds to a probe request or to an association request from a client.

There are other methods to get a hidden SSID. We could force the connected clients to reassociate with the AP sending them de-authentication packets, as we will see in \textit{Chapter 7, Wireless Client Attacks}. We could also analyze captured association and probe request/response packets with Wireshark to recover the SSID. We will cover packet dumping and analysis on \textit{Chapter 4, WEP Cracking} and \textit{Chapter 5, WPA/WPA2 Cracking}, about WEP and WPA/WPA2 cracking.

We can write the output to a file using the \texttt{-w} or \texttt{--write} options followed by the file name. Airodump-ng can save the output in various formats (pcap, ivs, csv, gps, kismet, and netxml), compatible with Kismet and packet analysis tools such as Wireshark.

Airodump-ng also allows to select specific channels through the \texttt{--channel} or \texttt{-c <ch_nr1,ch_nr2,...,ch_nrN>} option:

\begin{verbatim}
airodump-ng -c 1 -w output mon0
\end{verbatim}
Wireless scanning with Kismet

Kismet is a powerful passive scanner available for different platforms and is installed by default on Kali. It is not simply a scanner, but also a wireless frame analysis and intrusion detection tool.

Kismet is composed of two main processes: kismet_server and kismet_client. The kismet_server component does the job of capturing, logging, and decoding wireless frames. Its configuration file is kismet.conf and it is located at /etc/kismet/ on Kali Linux. The kismet_client frontend is a ncurses-based interface that displays the detected APs, statistics, and network details. To run it, we type kismet on the command line or navigate to Kali Linux | Wireless Attacks | 802.11 Wireless Tools | Kismet from the Application Menu:
As we can see, Kismet prompts us to start the server and we choose Yes and then Start in the following prompt. Then a message saying that no packet sources are defined could appear and we are asked to add a packet source:

```
INFO: No packet sources defined. You MUST ADD SOME using the Kismet client, or by placing them in the Kismet config file (/etc/kismet/kismet.conf)
INFO: Kismet server accepted connection from 127.0.0.1
```

The packet source is our monitor mode interface mon0 and we insert it in the Intf field in the subsequent prompt:
The packet source can also be set in the `kismet.conf` file, in the `ncsource` directive, as we can see in the following screenshot:

![Kismet configuration screenshot](image)

This is the recommended way to configure the packet source, avoiding to do it manually each time Kismet is started.
We close the server console and the client interface is displayed. To access the menu on the top of the window, we press the ~ key and move over the entries with the arrow keys. Kismet interface and behavior are customizable by navigating to Kismet | Preferences:

The screen is divided into the following main sections, from the top to the bottom: network list, client list, packet graph, status, and the general info side panel on the right. You can choose which sections to visualize in the View menu:
Chapter 3

The Network List shows the detected networks in the default Auto-fit mode.

To select a network and see its details and the clients connected, we need to change the sorting method to another, for example, using **Type** or **Channel** in the **Sort** menu. Then we can select a network on the list by clicking on it with the mouse:

Navigate to **Windows | Network Details** for more detailed information, such as the BSSID, the channel, the manufacturer, the signal level, packet rate, and so on:
If we select the **Clients** options, we can see the clients connected to the network, along with useful information such as the MAC address, the packets exchanged and the client device manufacturer.

In case of networks with cloaked SSID, Kismet shows the string `<Hidden SSID>` in place of the network name. When a client tries to connect to the network, the AP sends the SSID in clear in the response packets, allowing Kismet to uncover it, as we have already seen with Airodump-ng.

Kismet generates the following log files, by default in the directory from which it has been started (but we can change this in the `logtemplate` directive in `kismet.conf`):

- A packet capture file
- Networks in text format (.nettxt)
- Networks in XML format (.netxml)
- GPS data in XML format (.gpsxml)

The packet capture files can then be examined by Wireshark and can contain spectrum data, signal and noise levels, and GPS data.
Indeed, Kismet, as well as Airodump-ng, can be integrated with a GPS receiver, through the gpsd daemon, to establish the coordinates of the networks, which could also be used to realize graphical maps with apposite tools, such as GISKismet.

GISKismet
GISKismet is a visualization tool for Kismet, included by default in Kali Linux that allows to import the .netxml files into a SQLite database, so that we can execute SQL queries on it, and to build graphs and maps of the networks. This tool could be very useful especially when scanning large networks with many access points. For more information, see the GISKismet website http://trac.assembla.com/giskismet/wiki.

Summary
In this chapter, we introduced the IEEE 802.11 standard and the typical wireless LAN deployment in infrastructure mode. Then we covered the basic concepts of wireless scanning and saw how to practically discover and gather information about wireless networks, using two of the most effective tools included in Kali Linux: airodump-ng and Kismet.

In the next chapter, we will cover the WEP protocol, explaining why it is insecure, and see how it can be cracked using the tools provided with Kali Linux.
Where to buy this book

You can buy Kali Linux Wireless Penetration Testing Essentials from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.